Rodríguez de los Santos, G., Reviriego, P. & Hernández, J. A. (2015). Packet Coalescing Strategies for Energy Efficient High-Speed Communications Over Plastic Optical Fibers. *Journal of Optical Communications and Networking*, 7 (4), pp. 253-263.

DOI: 10.1364/JOCN.7.000253

Packet Coalescing Strategies for Energy Efficient High-Speed Communications Over Plastic Optical Fibers

Gerson Rodríguez de los Santos, Pedro Reviriego, José Alberto Hernández

Many recent standards for wireline communi-cations have included a low-power operation mode for energy efficiency purposes. The recently approved VDE standard 0885-763-1 for high-speed communication over plastic optical fibers has not been an exception. The low-power mode is used when there is no data to be transmitted over the line, thus making consumption more proportional to network load. Furthermore, packet coalescing has been proposed in the literature to minimize the transitions between the low-power and active modes, thus reducing the energy penalties associated with such transitions. This article proposes an adapted version of packet coalescing for the periodic structure of the VDE 0885-763-1 physical layer. Such an algorithm attempts to fulfill active periods of transmission with data, showing an improved energy efficiency over conventional packet coalescing strategies. This conclusion is evaluated via simulation with both synthetic Poissonian traffic and real traces.

Energy efficiency; Packet coalescing; Plastic optical fibers; VDE 0885-763-1.

I. Introduction

In the past few years, several new standards for wireline communications have included energy-efficiency features, mainly on attempts to reduce the overall power consumption and the carbon footprint of communication devices [1]. What is probably the most widely known example is IEEE standard 802.3az for energy efficient Ethernet (EEE) [2,3], whereby a low-power (also called "sleep") mode was introduced to allow energy savings when no data was pending for transmission. Many subsequent standards have followed this philosophy of introducing a low-power mode that can be used during idle periods of activity. This is also the case of the recently approved VDE standard 0885-763-1 for high-speed communication over plastic optical fibers (POFs) [4,5].

The use of POFs as a transmission medium is interesting due to their immunity to electrical interference, ease of installation, and reduced weight and cost [6]. Such positive features make POFs attractive for applications in automotive networks in which weight, cost, and interference are important, or for home networking in which installation and cost are critical [6,7]. Indeed, POFs are already extensively used in multimedia automotive applications [8].

VDE standard 0885-763-1 defines a configurable physical layer that supports different transmission speeds and link lengths. In this article, we focus on the VDE 0885-763-1 configuration for 1 Gb/s bidirectional communication for POF links of up to 50 m, but it is worth noting that the standard also allows a 100 Mb/s configuration mode for distances farther than 50 m, and even an adaptive rate mode that adjusts the speed to the channel quality conditions. The selection of the 1 Gb/s configuration is driven by the recent creation of a new IEEE 802.3 task force on gigabit Ethernet over POF [9]. This group aims to define a standard that will supersede the VDE 0885-763-1 one and ensure compatibility with Ethernet. To facilitate the process, VDE 0885-763-1 has been withdrawn, although there are compliant devices already on the market. In any case, for the new Ethernet standard, the use of the physical layer of VDE standard 0885-763-1 configured for 1 Gb/s is one of the candidate solutions. This means that the energy efficiency mechanisms studied in this paper may also be used in the future Ethernet standard.

Concerning energy efficiency in Ethernet (the so-called IEEE standard 802.3az), the low-power mode was introduced to allow Ethernet transceivers to save energy when no data was pending for transmission. Indeed, the standard was expected to improve the proportionality between power consumption and network load. However, as demonstrated in [10], the extra energy cost (energy overheads) of entering and exiting this low-power mode has a large energy penalty such that, if transitions occur too frequently, energy savings are reduced.

To minimize the number of low-power to active transitions and vice versa, the use of packet coalescing was further introduced in the literature [11]. The idea of packet coalescing is very simple: once the link has entered the low-power mode, it transitions back to the active mode only when a number of packets (or bytes) are ready for transmission, specified by the max-size parameter. This strategy greatly reduces the number of transitions and their associated energy overheads.

G. Rodríguez de los Santos (e-mail: gsantos@it.uc3m.es) and J. A. Hernández are with the Department of Telematic Engineering, ADSCOM Group, Universidad Carlos III de Madrid, Spain.

P. Reviriego is with the School of Engineering, Universidad Antonio de Nebrija, Madrid, Spain.

Clearly, implementing packet coalescing has an impact on packet delay, since a packet departure must wait until a number of other packets have arrived and the max-size criteria is met. To avoid excessively long delays, the max-size threshold criteria is combined with a max-delay threshold that forces the transition to the active mode and subsequent packet departure as soon as sufficient packets have arrived or the waiting delay of the first packet has reached the max-delay limit, whichever occurs first. Thus, fine tuning packet coalescing strategies comprises a trade-off between energy savings and network performance metrics measured in terms of packet delay [12–14].

The use of packet coalescing in VDE standard 0885-763-1 is analyzed throughout this article for the first time to the best of our knowledge. Basically, the periodic structure of the physical layer of VDE standard 0885-763-1 makes the packet coalescing algorithms used in EEE nonoptimal, as explained in further sections. Therefore, an adapted version of packet coalescing to VDE 0885-763-1 is designed, optimized, and evaluated both with synthetic traffic and real packet traces. The results show that packet coalescing can be particularly useful at low loads and especially when packets are short.

The remainder of this work is organized as follows. Section II reviews important aspects of both the VDE 0885-763-1 physical layer and previous work on energy efficiency, useful material to follow the rest of the paper. Section III introduces a number of packet coalescing strategies for VDE 0885-763-1, which are further evaluated and optimized in a number of scenarios in Section IV. Finally, Section V concludes this work with its main findings and some ideas for future work.

II. Background

A. Brief Review of VDE Standard 0885-763-1

The physical layer specified in VDE standard 0885-763-1 is based on the periodic frame structure shown in Fig. 1. This frame contains pilots, physical layer headers (PHSs), and codewords (CWs) [4]. Pilots (S1, S2) are used for physical layer functions like timing recovery and equalization. PHSs are used to exchange parameters between the edges, for instance, coefficient equalizers. Finally, the CWs are used to carry the user data bits along with extra bits for error correction. VDE standard 0885-763-1 uses multilevel coset coding (MLCC) [15] with three coding levels. Different Bose–Chaudhuri–Hocquenghem (BCH) codes are used for error correction in each level.

As shown in Fig. 1, each physical layer frame comprises 1 S1 and 13 S2 pilot sub-blocks, 14 physical header sub-blocks, and 112 CWs. Each codeword comprises 2016 symbols, while sub-block pilots and headers contain only 160 symbols.

Thus, each physical frame has a total of

$$112 \times 2016 + (1 + 13 + 14) * 160 = 230{,}272 \text{ symbols.}$$

When using the 1 Gb/s configuration, the symbol frequency is 312.5 MHz, thus the transmission of a frame requires 736.8704 μs. For this bit rate, a 16 pulse amplitude modulation (PAM) is used.

In a nutshell, CWs are transmitted in groups of four. Layer 2 data frames are encapsulated in these blocks of four CWs. If a Layer 2 data frame is too long, it is fragmented among four consecutive CW groups. Each group of four CWs is separated by a PHS/S header. Each group of four CW + PHS/S then takes

$$\frac{4 \times 2016 + 160 \text{ symbols}}{312.5 \text{ MHz}} = 26.3168 \text{ μs per group.}$$

At 1 Gb/s configuration, such a group can carry 26,316.8 Layer 2 bits (approximately 3290 bytes) as it follows from

$$26.3168 \text{ μs} \times 10^9 \text{ bit/s} = 26{,}316.8 \text{ bits} = 3289.6 \text{ bytes.}$$

This periodic structure of VDE 0885-763-1 is crucial for understanding one of the algorithms proposed in this paper. It is also worth mentioning that this periodic structure highly contrasts with those of other technologies like Ethernet. In this sense, because there is certain comparison with EEE for gigabit Ethernet throughout this paper, the interested reader has the opportunity to study the gigabit Ethernet physical level at a detailed level in [16]. In that case, there is no concept of periodic frame, and transmission can start or end at any point in time.

B. Previous Studies on Energy Efficiency in VDE 0885-763-1

VDE standard 0885-763-1 enables energy savings by stopping the transmission of groups of four CWs, as

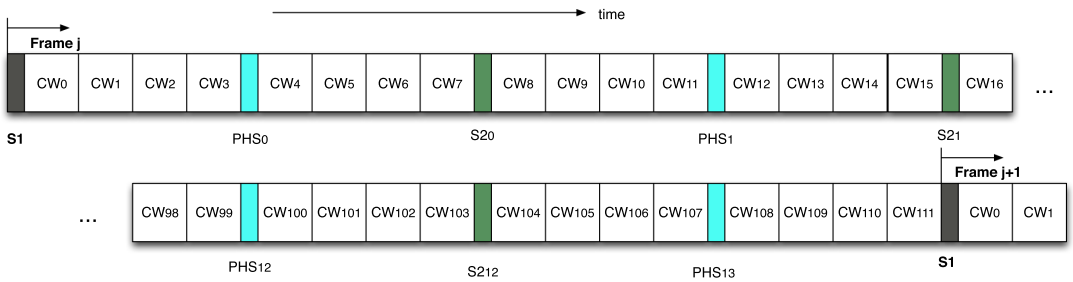


Fig. 1. Illustration of the frame structure in VDE standard 0885-763-1.

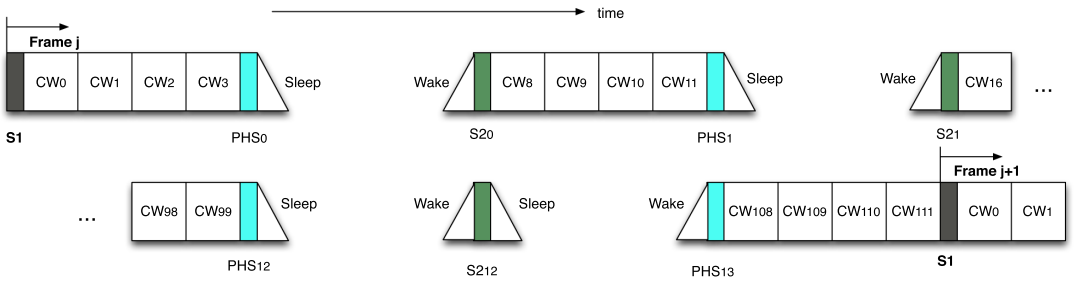


Fig. 2. Illustration of the use of the low-power mode defined in VDE standard 0885-763-1.

illustrated in Fig. 2. In the absence of data, the link enters the low-power mode and only the pilots and physical headers are transmitted to ensure that the transmitter and receiver are fully aligned and ready to begin data transmission as soon as new data arrives for transmission.

If a new packet arrives during the low-power mode, the standard states that the transceiver must wait until the next group of CWs to activate the link and dispatch such a packet.

This is rather different than the IEEE 802.3az EEE standard, whereby a given link may be activated or deactivated at any time. Furthermore, in EEE, the transition times from sleep to active (wake-up times T_w) and from active to sleep (sleep times T_s) are substantially large, thus producing an important energy penalty if many transitions between states occur (see [10]). This effect is particularly harmful for short packets and at low traffic loads.

However, in the VDE standard 0885-763-1, the energy overhead is mostly due to the transmission of wasted CWs, that is, CWs that carry no user data. For example, if a small 64-byte packet arrives, the link is activated for the whole group, that is, 26.3 μs, but only 0.5 μs are actually used in sending those 64 bytes at 1 Gb/s. In other words, activating a whole group just for the transmission of a short 64-byte packet has the following per-group *efficiency* η:

$$\eta_{64\,\text{byte}} = \frac{64}{3290} = 1.95\%.$$

The transmission of a large data packet (i.e., 1500 bytes) has a per-group efficiency of

$$\eta_{1500\,\text{byte}} = \frac{1500}{3290} = 46.59\%.$$

In order to achieve high efficiency values, it is desirable to achieve near 100% values of such per-group efficiency η, filling as much as possible full groups of four CWs.

The authors of [5] compared the energy consumption versus network load for a link using VDE standard 0885-763-1 at 1 Gb/s and a gigabit Ethernet link under the assumption of Poisson packet arrival times. The simulations in [5] showed that VDE was more efficient than EEE at 1 Gb/s (see Fig. 3, which is Fig. 6 in [5]).

Essentially, the per-group efficiency values are poor, especially at low loads, as demonstrated in the next example.

Example. Consider packet arrivals following a Poisson process with rate λ packets/s and average packet sizes of 600 bytes/packet. Let us assume a network load of $\rho = \lambda E(X) = 0.1$, thus,

$$\lambda = \frac{\rho}{E(X)} = 20{,}833.3 \text{ packets/s},$$

since $E(X)$ is the average service time per packet, that is,

$$E(X) = \frac{8.600 \text{ bit}}{10^9 \text{ bit/s}} = 4.8 \text{ μs}.$$

In such a scenario, the number of packet arrivals within an interval of length $T = 26.3168$ μs follows the Poisson distribution:

$$P(N(T) = k) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}, \qquad k = 0, 1, 2, \ldots.$$

When $\rho = 0.1$ (i.e., $\lambda T = \rho \frac{T}{E(X)} = 0.5483$):

$$\begin{aligned} P(N(T) = 0) &= e^{-0.5483} = 0.5780, \\ P(N(T) = 1) &= 0.5483 e^{-0.5483} = 0.3169, \\ P(N(T) = 2) &= \frac{0.5483^2}{2!} e^{-0.5483} = 0.0869, \\ P(N(T) \geq 3) &= 0.0183, \end{aligned} \tag{1}$$

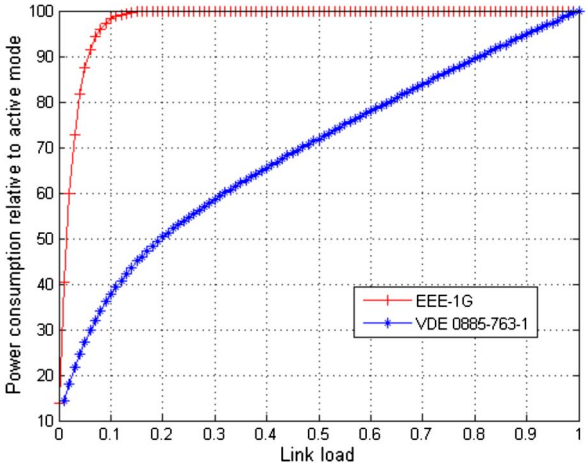


Fig. 3. Power consumption versus load for 600 byte packets: com-parison of EEE and VDE standard 0885-763 (from [5]).

which means that about 57.8% of the time, cycles do not carry any traffic and may be switched to the low-power mode, 31.7% of the time a cycle carries a single packet of length 4.8 μs (this results in a per-group efficiency 18.24%), 8.7% of the time the cycles carry two packets (per-group efficiency of 36.48%), and finally only 1.8% of the time, the cycle carries three packets or more.

Thus, the average efficiency when a group carries any data (one packet or more) can be approximated by

$$\begin{aligned}\bar{\eta} &\approx \sum_{k=1}^{\infty} \eta_k p(N(T)=k|N(T)\geq 1) \\ &= \sum_{k=1}^{\infty} \frac{kE(X)}{T} \frac{\frac{(\lambda T)^k}{k!}e^{-\lambda T}}{1-e^{-\lambda T}} \\ &= \frac{\rho}{1-e^{-\lambda T}}. \end{aligned} \tag{2}$$

This is an approximation since, at high loads, the transmission of the packets in the queue may require more than a single cycle. In our example, the average efficiency obtained is 23.7%.

This is why the energy efficiency mechanism defined for VDE 0885-763-1 is so inefficient at low loads under Poissonian traffic. Clearly, the way to improve energy efficiency in VDE 0885-763-1 comprises defining mechanisms to fill up cycles, that is, increasing the per-cycle efficiency. The next section reviews packet coalescing as a means to improve energy efficiency and its application to the particular physical layer of VDE 0885-763-1.

III. Packet Coalescing for VDE Standard 0885-763-1

A. Traditional Coalescing Algorithms

Traditional coalescing algorithms, such as those explained in [11], work as follows: When no data is pending for transmission, the link switches to the low-power mode. However, the link is not put back to the active mode when a single data frame arrives. Instead, the device waits until a number of s_c bytes have arrived or a timer t_w has expired, whichever occurs first.

For example, consider the hypothetical case of deterministic 1500-byte packet sizes and a coalescing policy specified by a data-limit value of $s_c = 3000$ bytes and a sufficiently large delay limit $t_w \to \infty$, such that packet coalescing is always triggered by size, never by delay.

The resulting per-group efficiency of such a packet coa-lescing algorithm would then be [see Fig. 4(a)]

$$\eta = \frac{3000}{3290} = 91.2\%,$$

which is close to the 100% target cycle efficiency.

On the other hand, a second packet coalescing algorithm with $s_c = 4500$ bytes is expected to have a poorer average cycle efficiency [see Fig. 4(b)],

$$\eta = \frac{1}{2}\left(\frac{3290}{3290} + \frac{4500–3290}{3290}\right) = 68.4\%,$$

since this algorithm manages to fill up one group and half of the next group, as observed in the figure.

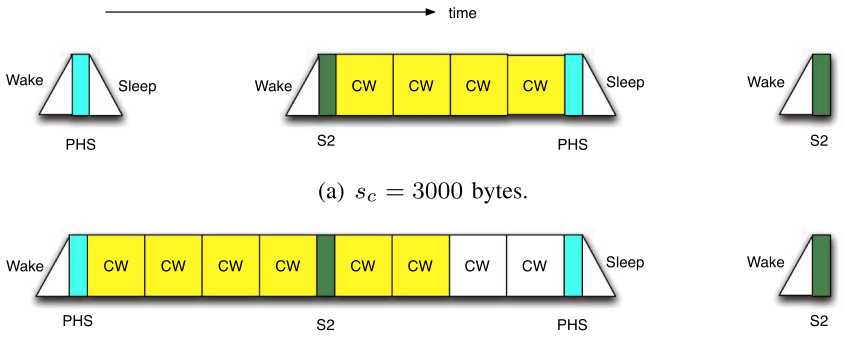
This behavior is rather different than that observed in packet coalescing for EEE, where the greater the size threshold, the better the efficiency achieved.

Consequently, the overall efficiency of a coalescing algorithm directly depends on its capability to fill groups of CWs entirely, that is, 3290 bytes. In this light, packet coalescing strategies need to be adapted to this behavior.

B. Coalescing Algorithm Proposals for VDE 0885-763-1

This subsection aims at comparing two different strategies to perform packet coalescing for VDE 0885-763-1, namely, *classical* and *strict cycle filling*. We assume the same size and delay thresholds in both of them.

1) Classical Coalescing: A detailed description of this algorithm is shown in Algorithm 1. Essentially, in this strategy, the waiting timer starts as soon as the first packet arrival occurs. Next, the algorithm waits until a number of



(a) $s_c = 3000$ bytes.

(b) $s_c = 4500$ bytes.

Fig. 4. Examples of transmission for different coalescing parameters.

s_c bytes have arrived or the maximum waiting-time threshold t_w has expired, whichever occurs first.

Algorithm 1 Detailed Description of the *Classic Coalescing* Strategy for VDE 0885-763-1

```
/* Check if there is an ongoing burst */
if burstongoing == true then
  transmitCurrentPacket();
  if endofburst == true then
    burstongoing = false;
  end if
  waitForNextEvent();
else
  if queue.timeout() == true then
    queue.sendAll();
    burstongoing = true;
    continuetx = true;
  else if queue.bytesInQueue() >= SCBYTES then
    queue.sendAll();
    burstongoing = true;
    continuetx = true;
  else if queue.bytesInQueue() > 0 and continuetx ==
  true then
    queue.sendAll();
    burstongoing = true;
  else
    continuetx = false;
  end if
    waitForNextEvent();
end if
/* continuetx will be set to false too if the transceiver goes to
sleep due to inactivity */
```

When this happens, all packets in the buffer are transmitted at the beginning of the next cycle. In addition, all packet arrivals during the transmission of these bytes are also transmitted. When the buffer completely empties, the algorithm restarts itself. If new packets arrive during an active cycle after the queue has emptied, these packets have to wait until the s_c and t_w requirements are met again.

This behavior attempts to imitate the coalescing algorithm defined for EEE, whereby the transceiver is put in the low-power mode after transmitting all existing packets in the queue. In this light, it is worth noting that, although the algorithm defines a maximum transmission size threshold of s_c bytes, in fact more than s_c bytes may be transmitted per active cycle, since more packets may arrive during the transmission of other previously buffered packets. This is best illustrated in the example of Fig. 5.

2) Strict Cycle-Filling Coalescing: A detailed description of this algorithm can be seen in Algorithm 2. In this case, the algorithm behaves similarly to the classical coalescing strategy, but differs in the fact that it transmits only the minimum number of packets to achieve at least s_c bytes *strictly*. Further packet arrivals are transmitted only if they fit in the remaining space of the active cycle. Once such an active cycle has finished, the algorithm is reset.

Algorithm 2 Detailed Description of the *Strict Cycle-Filling Coalescing* Strategy for VDE 0885-763-1

```
/* Check if there is an ongoing burst */
if burstongoing == true then
  transmitCurrentPacket();
  if endofburst == true then
    burstongoing = false;
  end if
  waitForNextEvent();
else
  if queue.timeout() == true then
    if queue.bytesInQueue() >= SCBYTES then
      queue.sendAtLeast(SCBYTES);
    else
      queue.sendAll();
    end if
    burstongoing = true;
    continuetx = true;
  else if queue.bytesInQueue() >= SCBYTES then
    queue.sendAtLeast(SCBYTES);
    burstongoing = true;
    continuetx = true;
  else if queue.bytesInQueue() > 0 and continuetx ==
  true then
    if queue.packetsFitInCycle() == true then
    queue.SendPacketsThatFitInCycle();
    burstongoing = true;
    end if
  end if
  waitForNextEvent();
end if
/* continuetx will be set to false if the transceiver goes to sleep
due to inactivity */
```

This algorithm aims at fitting the slotted nature of VDE standard 0885-763-1. In this light, the s_c threshold value must be chosen to fill up most of the cycle, that is, close to 3290 bytes.

Example. To illustrate the behavior of both classic and cycle-filling packet coalescing strategies, consider the example of Fig. 5, where cycles are normalized to unity length, and packet sizes are smaller than one. We consider both coalescing strategies with configuration $s_c = 0.75$ and $t_w \to \infty$. For simplicity, the control packet interval length is considered negligible.

In this example, nine packet arrivals, P_1 through P_9 at times t_1 to t_9, respectively, are considered. Packets P_1 and P_2 arrive at times t_1 and t_2 during a low-power cycle. Because the s_c threshold is crossed, the next cycle becomes active and both packets are transmitted.

P_3, of length 0.1, is immediately transmitted after P_2 in both algorithms, but for different reasons. The classic algorithm states that any packet arrival before the queue empties must be transmitted. The strict cycle-filling algorithm states that new packet arrivals can only be transmitted if they fit in the remaining space of the active cycle; this is the case for P_3.

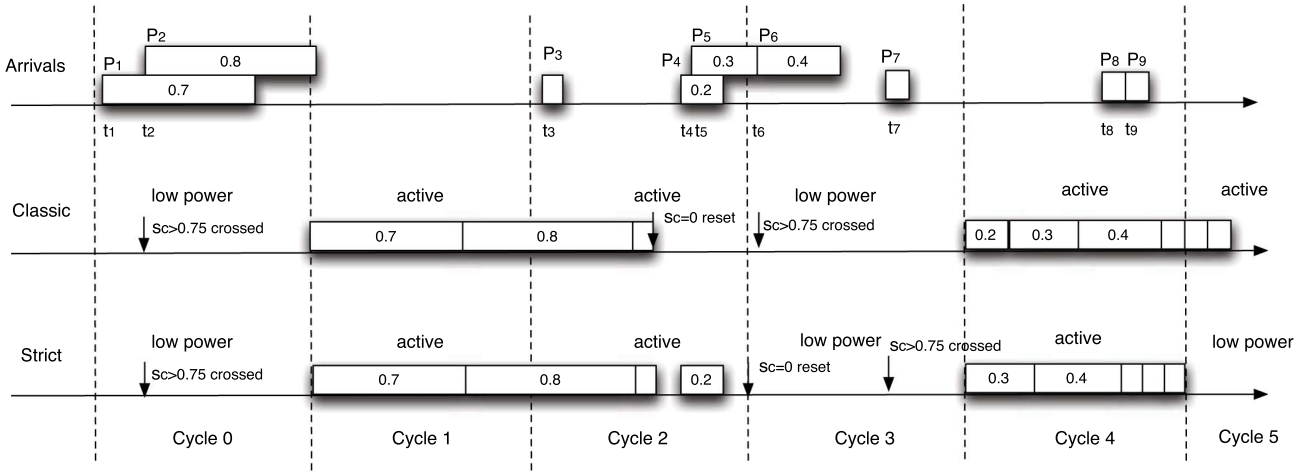


Fig. 5. Differences between classic and cycle-filling packet coalescing strategies, $s_c = 0.75$.

After the transmission of P_3 the queue empties, and Algorithm 1 restarts itself, but Algorithm 2 does not. For this reason, P_4 can be transmitted during cycle 2 for Algorithm 2, but has to wait for Algorithm 1 until the $s_c$condition is met again. In contrast, P_5 is not transmitted after P_4 since it does not fit in the remaining space of cycle 2.

Cycle 3 is a low-power one, but cycle 4 is again active since the s_c thresholds have been crossed during cycle 3.

As shown, the classic coalescing algorithm uses two active cycles (cycles 4 and 5) since new packets (P_8 and P_9) arrive before the queue empties. In contrast, the strict algorithm manages to fit all packets within one cycle. It is worth remarking that Algorithm 2 only transmits more packets than s_c bytes if these fit in the remaining space of an already active cycle.

The average cycle efficiencies in both cases are

$$\eta_{Alg.1} = \frac{1}{4}(0.7 + 0.8 + 0.1 + 0.2 + \cdots + 0.3 + 0.4 + 0.1 + 0.1 + 0.1) = 70.0\%,$$

$$\eta_{Alg.2} = \frac{1}{3}(0.7 + 0.8 + 0.1 + 0.2 + \cdots + 0.3 + 0.4 + 0.1 + 0.1 + 0.1) = 93.3\%,$$

As shown, the cycle-filling algorithm is expected to achieve higher cycle efficiencies than the classic one. Obviously, the example of Fig. 5 is not a proper characterization of the algorithm, since just a specific case is being presented. However, it illustrates the main differences between the two coalescing algorithms. A statistically relevant evaluation is conducted in the next sections to effectively confirm such an intuition. The next section further evaluates both efficiency and packet delay due to coalescing for both algorithms under different network conditions and scenarios.

IV. Evaluation

As existing devices that implement VDE standard 0885-763-1 do not yet support the energy efficient mode, the evaluation has been done by simulation. This was also the case for the initial studies on EEE [10–14] that were subsequently validated with real measurements once devices that supported EEE became available. As for EEE, both synthetic traffic and packet traces are used in the simulations presented in this section.

To obtain the results of this article, a custom-made simulator has been created. It can be downloaded through the website in [17]. The packet traces used are also available on the website.

A. Synthetic Poisson Traffic: Energy Performance

This first set of experiments aims at evaluating the behavior of the two packet coalescing algorithms (namely, classical and strict cycle-filling) under synthetic Poissonian-like traffic. In this light, we have considered two different packet sizes: short 64-byte packets and long 1500-byte packets, which translate into service times of 0.512 and 12 μs, respectively.

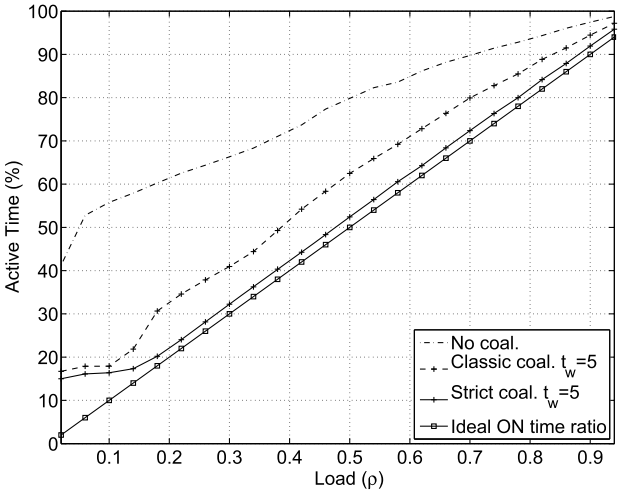
Packet interarrival times are then exponentially distributed with mean $1/\lambda$, where λ is calculated, for an average service time $E(X)$, as

$$\lambda = \frac{\rho}{E(X)}$$

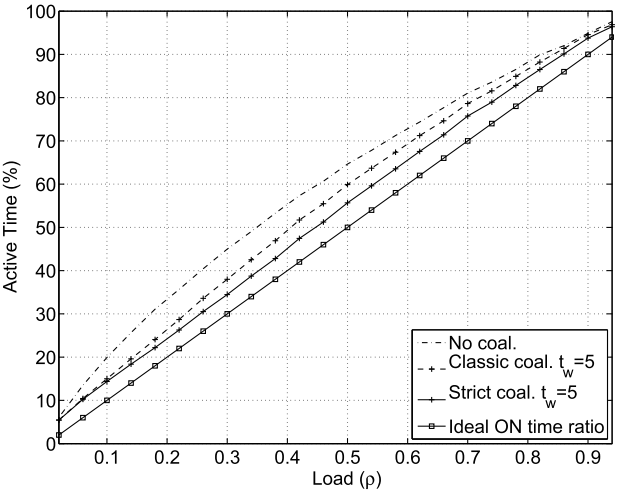
for different load values of $\rho \in (0, 1)$.

Figure 6 shows the percentage of cycles that are active cycles (percent of time ON) of the two packet coalescing strategies along with the ideal case of proportional consumption load and the case where no packet coalescing strategy is employed.

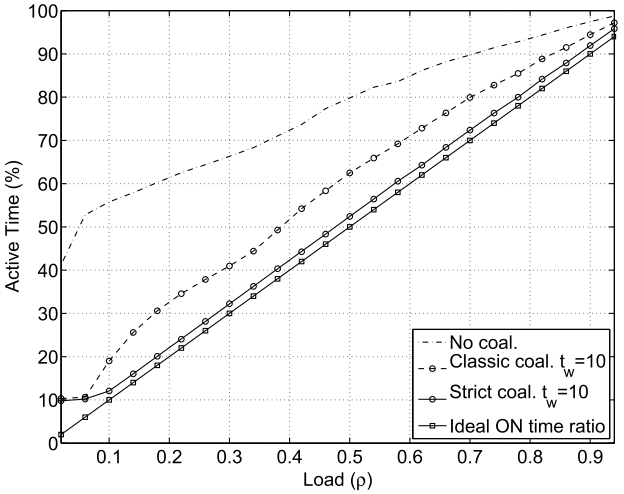
In such experiments, the packet coalescing strategies have been configured to follow: $s_c = 3000$ bytes (a bit less than a full cycle) and $t_w = \{5, 10, 25\}$ cycles of maximum waiting time (that is, 161.5, 263.2, and 657.9 μs, respectively, since each cycle lasts for about 26.3 μs).



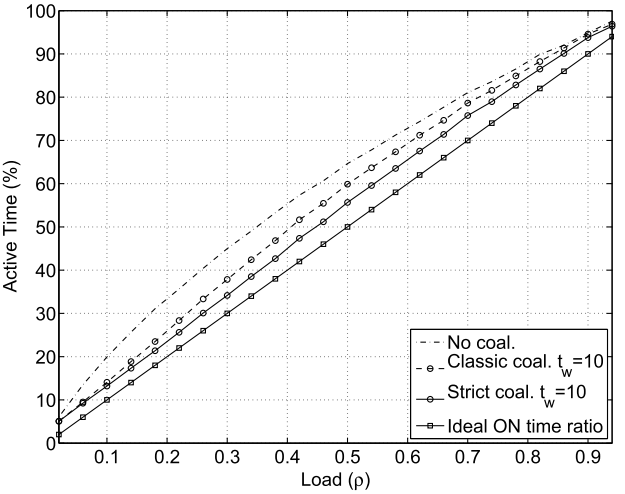
(a) Short packets, $t_w = 5$.



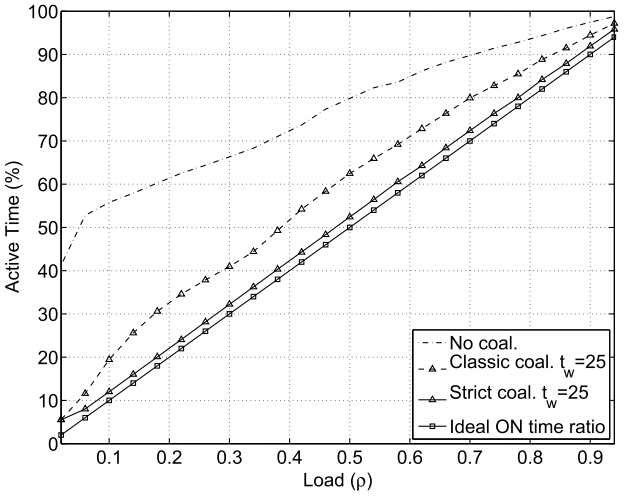
(b) Long packets, $t_w = 5$.



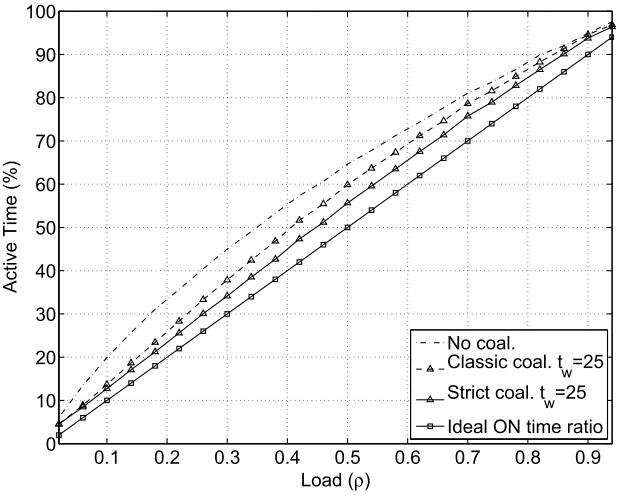
(c) Short packets, $t_w = 10$.



(d) Long packets, $t_w = 10$.



(e) Short packets, $t_w = 25$.



(f) Long packets, $t_w = 25$.

Fig. 6. Percentage of active time versus load for different packet lengths and t_w.

In a nutshell, the results observed reveal that

1) employing any coalescing algorithm clearly outperforms energy efficiency without coalescing, especially when packets are short, and
2) the t_w parameter has a moderate impact on the energy consumption results. Only at low loads do large values of t_w improve the energy-consumption figures. However, such small improvement at low loads is at the expense of larger average delay values, as shown in the next section.

The strict cycle-filling coalescing algorithm is closer to the ideal proportional consumption-load straight line than the classical coalescing algorithm in all cases. When packet sizes are large, the difference between the algorithms and the ideal linear case is not significantly important, but still the strict cycle-filling algorithm outperforms over the rest. Indeed, the strict cycle-filling manages to transmit all packets using a smaller number of active cycles, since it fills cycles better, leading to a higher cycle efficiency.

This is clearly observed in Fig. 7, where the average cycle efficiency is depicted. As shown, the cycle-filling coalescing algorithm approaches 100% cycle utilization when packets are short, and about 90% when packets are long, at loads greater than 20%, i.e., $\rho > 0.2$.

B. Synthetic Poisson Traffic: Delay Analysis

Finally, Fig. 8 shows the average delay experienced by packets under the two strategies, at different traffic loads ρ and for different values of t_w, i.e., 5, 10, and 25 cycles, that is, 263.17 and 657.92 μs. The size threshold is fixed to $s_c = 3000$ bytes.

Essentially, both coalescing strategies impose a delay penalty over the case where no coalescing is employed. However, in most cases, the strict cycle-filling strategy penalizes delay more than the classical coalescing. The reason behind that is that the classical coalescing algorithm has a number of cases where packets are transmitted earlier than in the cycle-filling case, although this behavior is the opposite in a few cases. This feature allows the classical coalescing case to perform better in terms of delay, but worse in terms of efficiency.

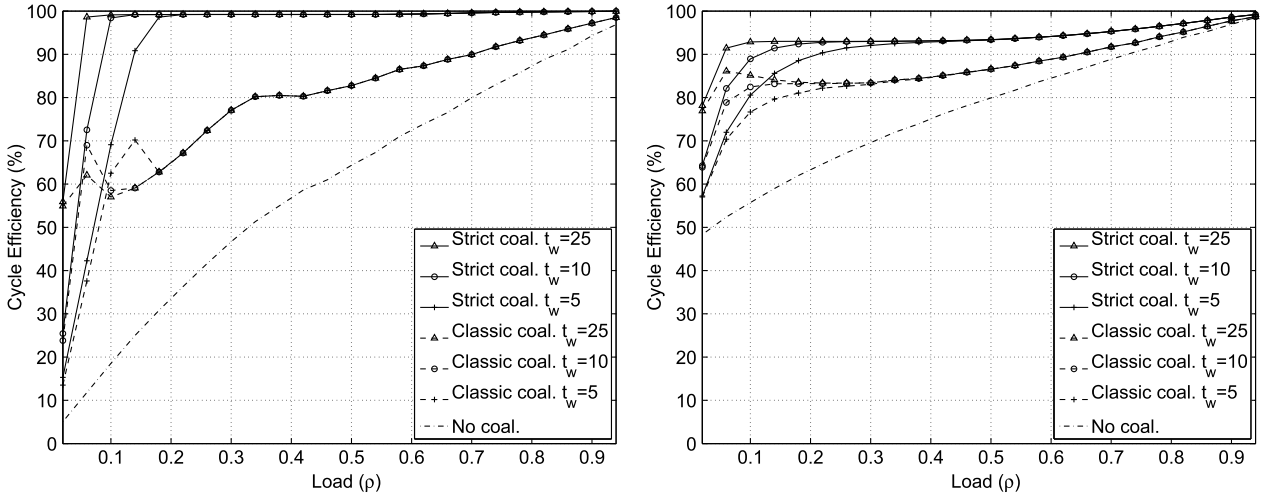
In addition to the average delay, Fig. 9 shows the maximum delay observed by packets under the two coalescing strategies for cases with short and long packets. As shown, the maximum delay is mostly influenced by the maximum waiting time parameter of the algorithm, that is t_w. At medium and high loads, the effect of t_w is negligible, since the algorithms are mostly triggered by the size constraint s_c.

C. Experiments With Real Traffic Traces at 1 Gb/s

This section evaluates the performance of the coalescing algorithms with real 1 Gb/s traces collected from a number of different scenarios. The traces collected contain packet arrival and packet service times, thus allowing estimation of the behavior of the different coalescing strategies. Different performance metrics have been computed for each monitored trace, as shown next.

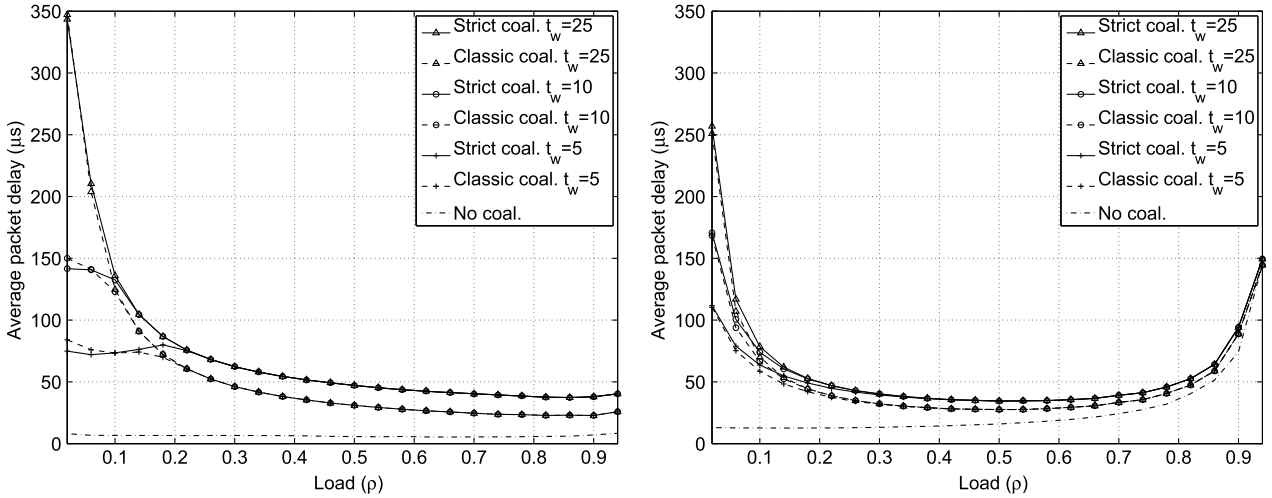
The first scenario under study considers a residential user under two typical cases: 1) YouTube video streaming and 2) BitTorrent file sharing. This user is connected by a 1 Gb/s Ethernet link to his/her HFC access (50/5 Mbit/s asymmetric down/upstream). The second scenario considers a university access link (1 Gb/s capacity) with highly multiplexed Internet traffic from students and staff members. Finally, the third scenario considers a number of server traces collected from Google's data centers. The traces comprise three typical server types: a file server, which is also involved in search queries, a second server devoted to only search queries, and a third one that acts as both file and application server. All these servers are connected via gigabit Ethernet.

Table I shows the cycle efficiency values η obtained under the two coalescing strategies, with different t_w



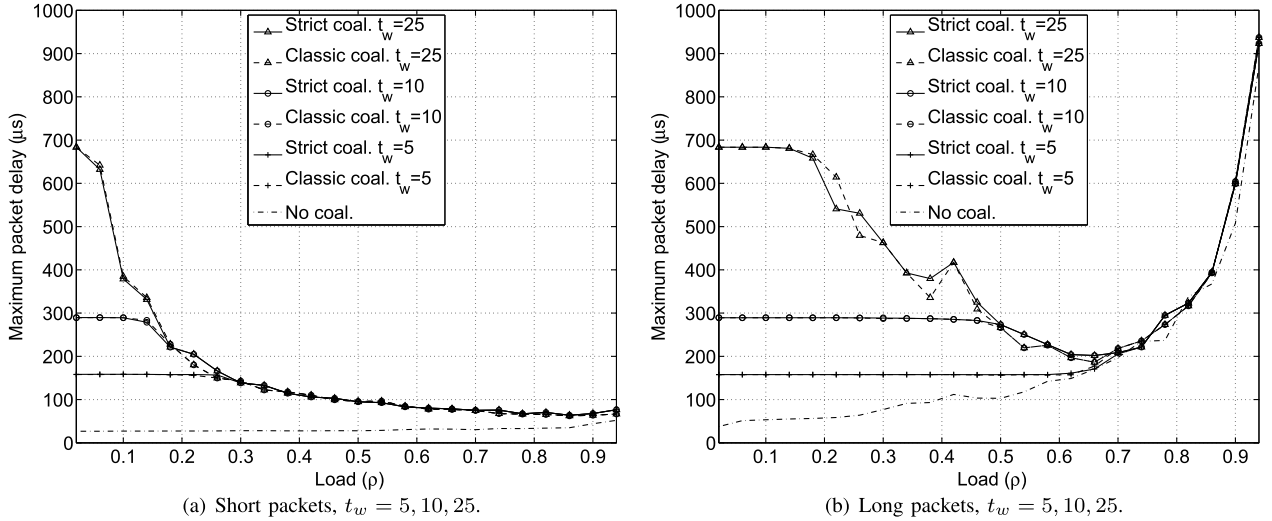
(a) Short packets. (b) Long packets.

Fig. 7. Average cycle efficiency with different packet coalescing strategies.



(a) Short packets, $t_w = 5, 10, 25$. (b) Long packets, $t_w = 5, 10, 25$.

Fig. 8. Average packet delay with different packet coalescing strategies.



(a) Short packets, $t_w = 5, 10, 25$. (b) Long packets, $t_w = 5, 10, 25$.

Fig. 9. Max packet delay with different packet coalescing strategies.

TABLE I
EXPERIMENTS WITH REAL TRAFFIC TRACES[a]

				$t_w = 5$		$t_w = 10$		$t_w = 25$	
Scenario	Direction	Link Load	No Coal	Classic	CycFil	Classic	CycFil	Classic	CycFil
BitTorrent	Input	0.61	18.39	21.58	22.50	24.95	25.41	30.84	31.64
BitTorrent	Output	1.75	36.61	41.30	41.85	49.56	50.12	60.17	60.99
YouTube	Input	0.0067	2.33	2.35	2.42	2.55	2.55	2.87	2.92
YouTube	Output	0.21	12.50	13.08	13.32	15.16	15.48	16.18	16.73
University	Input	10.79	40.73	59.80	65.14	69.23	75.44	75.94	82.58
University	Output	17.43	49.31	73.24	79.09	75.71	81.62	76.50	82.39
Data Center 1	Input	1.21	5.78	13.08	14.21	20.92	22.08	41.25	42.75
Data Center 1	Output	52.15	92.77	93.28	93.58	93.80	94.21	94.49	94.95
Data Center 2	Input	8.52	58.64	64.99	65.71	69.71	70.28	76.96	77.64
Data Center 2	Output	7.25	45.78	54.45	55.60	59.85	60.72	68.11	68.72
Data Center 3	Input	0.65	5.32	8.26	8.84	11.23	11.71	19.00	19.42
Data Center 3	Output	4.03	30.19	37.84	39.08	43.98	44.91	55.97	56.53

[a]Link load (%) and average cycle efficiency (%) for the classic and cycle-filling algorithms with different t_w values.

TABLE II
EXPERIMENTS WITH REAL TRAFFIC TRACES[a]

Scenario	Direction	Link Load	No Coal	$t_w = 5$		$t_w = 10$		$t_w = 25$	
				Classic	CycFil	Classic	CycFil	Classic	CycFil
BitTorrent	Input	0.61	12.32	106.99	104.91	181.86	182.92	386.08	387.89
BitTorrent	Output	1.75	11.98	117.60	116.89	183.92	184.01	324.90	326.98
YouTube	Input	0.0067	11.75	140.37	123.98	231.28	230.34	476.33	467.73
YouTube	Output	0.21	13.95	77.24	81.63	92.59	104.08	121.27	156.60
University	Input	10.79	12.91	69.41	73.31	99.54	106.53	154.21	163.32
University	Output	17.43	12.10	58.21	62.79	68.45	73.21	77.28	81.76
Data Center 1	Input	1.21	7.56	87.67	80.47	153.51	146.29	349.23	346.47
Data Center 1	Output	52.15	137.82	140.37	144.16	141.05	146.31	142.87	148.48
Data Center 2	Input	8.52	61.71	101.05	101.76	127.59	129.51	205.34	209.72
Data Center 2	Output	7.25	224.24	286.78	283.35	335.87	332.41	475.28	474.15
Data Center 3	Input	0.65	10.21	102.70	95.01	180.73	173.48	398.13	392.13
Data Center 3	Output	4.03	171.72	254.91	250.29	324.57	318.89	508.14	503.95

[a]Link load (%) and average delays (μs) for the classic and cycle-filling algorithms with different t_w values.

parameters. Table II shows the average delay obtained for the two coalescing algorithms.

As observed, employing any coalescing strategy significantly improves the cycle efficiency in all experiments. In particular, the strict cycle-filling performs better than the classical coalescing strategy, especially at medium to high loads and when packet sizes are small. In addition, the value of t_w plays a critical role in both energy savings and average delay experienced by packets.

In the first scenario, for the residential user, we observe that the link load is very small (up to 2% in the downlink direction, BitTorrent case). In such a case, important energy savings are achieved only when using a significantly large value of t_w, as observed in the last column. However, in such cases, the average delay due to coalescing is also significantly increased.

In the second scenario, the university access link, the difference in energy performance between the strategies is significantly large. As observed, the cycle efficiency goes from 40.73% without coalescing, up to 82.58% for strict cycle-filling coalescing with $t_w = 25$ in the input case, and from 49.31% to 82.39% in the output case. This translates into large energy savings.

Finally, in the third scenario, the cycle efficiency improvements differ from one dataset to another.

However, the behavior observed is consistent with the previous findings: 1) strict cycle-filling outperforms over classical coalescing; 2) the value of t_w plays a critical role in the cycle-filling efficiencies; 3) especially at low loads; and 4) for long packets.

Concerning average coalescing delay, the results are consistent with those observed from the synthetic Poisson traffic traces. Essentially, the two algorithms show very similar average delays; in some cases, the classic algorithm shows smaller delays and vice versa. We believe that the average packet size plays an important role in the differences between the two algorithms. The detailed analysis of these small differences is left for future work.

Take for instance the example of the Data Center 1 trace. In this trace, most packets (around 88%) have a size of 66 bytes in the input direction, while the most common size (97% of the packets) in the output direction is 1514 bytes. The average delay in the input direction is smaller for the strict cycle-filling algorithm than for the classic algorithm. The reason for this has to do with the fact that the strict cycle-filling algorithm allows adding new packet arrivals after the queue has emptied only if they fit in the remaining space of the already active cycle. This is obviously easier when packets are small, as is the case in the input direction.

V. SUMMARY AND DISCUSSION

This work has proposed two packet coalescing strategies to further improve energy efficiency in high-speed communications over POFs following VDE standard 0885-763-1. Both strategies, namely, classical and strict cycle-filling, attempt to fill up the active 26 μs cycles defined as the minimum active periods for data transmission. Each strategy has its pros and cons: while the cycle-filling strategy prioritizes the improvement of cycle efficiency, in some cases, it imposes more delay on individual packets than the classical coalescing algorithm.

These conclusions have been validated with both synthetic Poisson traffic and real traffic traces collected from a number of 1 Gb/s scenarios. Indeed, the experiments have revealed that the strict cycle-filling coalescing algorithm shows significantly better energy performance, especially at low loads and when the average packet size is large. Furthermore, the s_c and t_w parameters have been shown to clearly impact the two performance metrics of the coalescing algorithms, namely, energy efficiency and coalescing delay.

Future studies may involve profiling different scenarios and investigating the influence of the coalescing algorithms on different traffic patterns in terms of energy efficiency and coalescing delay. In addition, future work shall study how to dynamically adapt the s_c and t_w parameters

of the coalescing algorithm to changes in the traffic conditions, mainly load, average size, and interarrival times of packets. Moreover, further work should address the scalability of the algorithms proposed to different physical layer structures and standards as well as higher traffic rates than those of the study, 1 Gbps. Once transceivers that implement the energy efficiency features become available, future work should also involve implementing our algorithms in an actual POF link to measure the real energy savings that could be achieved.

Acknowledgments

The authors would like to thank K. Fu, G. Chesson, L. A. Barroso, and U. Holzle from Google for the anonymous traces from their data centers used in the experiment section. The authors would also like to acknowledge the Spanish-funded CRAMNET project (grant number TEC2012-38362-C03-01) for its support to the development of this work.

References

[1] K. J. Christensen, C. Gunaratne, B. Nordman, and A. D. George, "The next frontier for communications networks: Power management," *Comput. Commun.*, vol. 27, no. 18, pp. 1758–1770, Dec. 2004.

[2] "Energy efficient Ethernet," IEEE standard 802.3az, 2010.

[3] K. Christensen, P. Reviriego, B. Nordman, M. Benett, M. Mostowfi, and J. A. Maestro, "IEEE 802.3az: The road to energy efficient Ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, 2010.

[4] "Physical layer parameters and specification for high speed operation over plastic optical fibres type HS-BASE-P," VDE 0885-763-1 standard, Oct. 2013.

[5] P. Reviriego, P. Pérez de Aranda, and C. Pardo, "Introducing energy efficiency in VDE 0885-763-1 standard for high speed communication over plastic optical fibers," *IEEE Commun. Mag.*, vol. 51, no. 8, pp. 97–102, Aug. 2013.

[6] P. Polishuk, "Plastic optical fibers branch out," *IEEE Commun. Mag.*, vol. 44, no. 9, pp. 140–148, Sept. 2006.

[7] I. Mollers, D. Jager, R. Gaudino, A. Nocivelli, H. Kragl, O. Ziemann, N. Weber, T. Koonen, C. Lezzi, A. Bluschke, and S. Randel, "Plastic optical fiber technology for reliable home networking: Overview and results of the EU project POF-ALL," *IEEE Commun. Mag.*, vol. 47, no. 8, pp. 58–68, Aug. 2009.

[8] A. Grzemba, *MOST: The Automotive Multimedia Network from MOST25 to MOST150*. Franzis, 2008.

[9] "Gigabit Ethernet over plastic optical fiber task force," IEEE standard 802.3, 2015 [Online]. Available: http://www.ieee802 .org/3/bv/index.html.

[10] P. Reviriego, J. A. Hernández, D. Larrabeiti, and J. A. Maestro, "Performance evaluation of energy efficient Ethernet," *IEEE Commun. Lett.*, vol. 13, no. 9, pp. 697–699, Sept. 2009.

[11] P. Reviriego, J. A. Hernández, D. Larrabeiti, and J. A. Maestro, "Burst transmission in energy efficient Ethernet," *IEEE Internet Comput.*, vol. 14, no. 4, pp. 50–57, July/Aug. 2010.

[12] J. Meng, F. Ren, W. Jiang, and C. Lin, "Modeling and understanding burst transmission algorithms for energy efficient Ethernet," in *Proc. IEEE/ACM 21st Int. Symp. on Quality of Service (IWQoS)*, Montreal, June 2013.

[13] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "Bounded energy consumption with dynamic packet coalescing," in *Proc. IEEE NOC*, Vilanova i la Geltru, Spain, June 2012.

[14] A. Chatzipapas and V. Mancuso, "Modelling and real-trace-based evaluation of static and dynamic coalescing for energy efficient Ethernet," in *Proc. Int. Conf. on Future Energy Systems (e-Energy)*, Berkeley, CA, 2013, pp. 161–172.

[15] G. D. Forney, M. D. Trott, and S. Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 820–850, May 2000.

[16] M. Hatamian, O. E. Agazzi, J. Creigh, H. Samueli, A. J. Castellano, D. Kruse, A. Madisetti, N. Yousefi, K. Bult, P. Pai, M. Wakayama, M. M. McConnell, and M. Colombatto, "Design considerations for gigabit Ethernet 1000Base-T twisted pair transceivers," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1998, pp. 335–342.

[17] G. Rodriguez de los Santos Lopez, "POF simulator v0.1," ADSCOM Research Group, Department of Telematic Engineering, Universidad Carlos III de Madrid, Leganes, Madrid, Spain, 2014 [Online]. Available: http://www.it.uc3m.es/gsantos/pofsimulator.html.